

Stochastic level-set method for shape optimisation

Lester O. Hedges,^{1,2} H. Alicia Kim,^{3,2} and Robert L. Jack¹

¹*Department of Physics, University of Bath, Bath BA2 7AY, United Kingdom*

²*School of Engineering, Cardiff University, Cardiff CF24 3AA, United Kingdom*

³*Structural Engineering Department, University of California, San Diego, La Jolla, CA 92093 USA*

We present a new method for stochastic shape optimisation of engineering structures. The method generalises an existing deterministic scheme, in which the structure is represented and evolved by a level-set method coupled with mathematical programming. The stochastic element of the algorithm is built on the methods of statistical mechanics and is designed so that the system explores a Boltzmann-Gibbs distribution of structures. In non-convex optimisation problems, the deterministic algorithm can get trapped in local optima: the stochastic generalisation enables sampling of multiple local optima, which aids the search for the globally-optimal structure. The method is demonstrated for several simple geometrical problems, and a proof-of-principle calculation is shown for a simple engineering structure.

I. INTRODUCTION

Structural optimisation aims to provide performance improvements and/or weight savings by formulating an engineering structural design problem as constrained optimisation. The class of structural optimisation that is of interest in this manuscript is shape optimisation using the level-set method, which systematically modifies the structural boundary, i.e. design shape, to maximise or minimise the given performance metric while satisfying one or more constraints. A related class of structural optimisation is topology optimisation in which shapes and the number of boundaries are optimised. Topology optimisation is considered the most generic form of structural optimisation since the optimal solution is the most independent of the initial solution and can offer substantial performance improvements via unintuitive and creative designs [6, 19]. To this extent, shape optimisation is an important class of structural optimisation, critically enabling topology optimisation via the level-set method. It is noted that the level-set method naturally splits or merges boundaries thus topology optimisation is inherently enabled, although for the careful investigations presented in this article, we focus our attention to shape optimisation of a single external boundary of the structure.

It is well-known that many of the relevant applications involve optimisation problems that are *non-convex* – they support multiple locally-optimal designs, which correspond to local minima of the objective function. These local optima might be associated with different structural topologies, conflicting design requirements and/or with numerical aspects of the (discretised) computational problem. Examples of non-convex design spaces in the engineering literature are stress constrained optimization [3] and coupled multiphysics optimization [10]. Such systems cause problems for conventional (deterministic) optimisation schemes, which tend to converge to *local* optima, but miss the globally-optimal structure. The purpose of this paper is to exploit an analogy between such engineering problems and statistical mechanical systems with non-convex potential energy surfaces. This moti-

vates us to introduce a stochastic algorithm for exploring the space of possible structures, in order to sample multiple local optima and – eventually – converge to the global optimum.

In the engineering context, non-convexity and multiple optima present interesting dilemmas. If optimisation algorithms yield locally optimal structures which are much worse than the global optimum, they would not be considered useful design methods. However, generating multiple locally-optimal solutions can also provide multiple design ideas to engineers, offering a range of possible solutions with similar values of the objective function. In this case, an engineer may wish to consider several of these solutions, based on practical design requirements such as ease of manufacturing. This particularly true when there are several designs of similar objective function and constraint values, as in [3].

A common approach to topology optimisation is to employ gradient-based nonlinear programming, which can be applied to typical engineering design problems with $10^4 - 10^6$ design variables [2, 8]. In this case, iteration of the optimiser leads to one local optimum solution. Alternative (stochastic) approaches such as evolutionary algorithms, particle swarm optimisation and simulated annealing are capable of searching for multiple potential solutions and they have been applied to topology optimisation [14, 25, 26]. However the success of such methods has been limited, partly because they do not typically take advantage of information about the gradient of the objective function. Interested readers are referred to a critical review [20] which presents an example: Topology optimisation was applied to a small problem with just 144 variables, which was solved using the non-gradient method of differential evolution. This required 15,730 function evaluations. In contrast, a gradient-based topology optimisation method – Solid Isotropic Material with Penalisation (SIMP) – converged to a slightly superior solution after just 60 function evaluations. This motivates the research question of how to explore non-convex engineering design spaces effectively and efficiently.

One approach which can be successful in non-convex optimisation problems is to start with a deterministic

optimisation procedure and to add a stochastic component, so that the objective function can both increase and decrease as the algorithm runs. In this article, we introduce a stochastic optimisation procedure which aims to generate designs according to a prescribed distribution, analogous to the Gibbs-Boltzmann distribution of statistical mechanics. The method is built on the deterministic gradient-based topology optimisation method of Dunning and Kim [8, 21] and that method is recovered in the zero-noise limit of the stochastic process – this means that the stochastic method should perform at least as effectively as the deterministic one. Moreover, since the model is based on the Boltzmann distribution, we expect that it can be combined with parallel-tempering methods [11, 18, 23], which offer a systematic approach for exploring a range of near-optimal structures.

Our new method is based the level-set topology optimization (LSTO) method of Refs. [8, 21]. Unlike traditional topology optimisation (e.g. SIMP) in which the design variables (for example n) indicate whether each element of the design domain should exist ($n = 1$) or be absent ($n = 0$), the level-set method employs an implicit functional representation ϕ that directly represents the domain boundaries. A key advantage of this method is that topological changes in the shape of the object (for example the removal of holes) are associated with singularities in the behaviour of the boundary of the object, but do not involve any singularities in ϕ . See [17] for a description of the level-set method and its key advantages. The effect of the level-set representation is that topology optimisation can be reformulated as an extended shape optimisation, where boundary shapes are optimised and the number of boundaries can change. In this paper, we will focus on the shape optimisation aspect of the algorithm, where the level set method moves the boundaries using an advection equation. Within this scheme, the LSTO method corresponds to steepest descent of the objective function, which ensures robust convergence of the method to a locally-optimal design. We will show how this method can be extended to a stochastic method that can explore non-convex design landscapes.

This article presents several new results. In Sec. II, we review the method of [8], and we introduce some simplifications to that method, which clarify the relationship between that method and steepest descent optimisation of the objective function. In Sec. III, we introduce a stochastic process that explores a range of structures, parameterised by a noise strength T that is analogous to the temperature in statistical physics. The process is defined by a stochastic differential equation, analogous to Langevin equations in physics. Sec. IV includes several examples of the application of the stochastic method, including matching of a shape to a fixed design, and compliance minimisation of a simple two-dimensional engineering structure, subject to a constraint on its area. We discuss the nature of the shapes/structures explored by the stochastic method, and discuss the potential use of the method for practical optimisation of engineering

structures. Sec. V summarises our conclusions.

II. THE LEVEL SET TOPOLOGY OPTIMISATION METHOD

The optimisation problem considered here is the determination of a structural domain Ω that minimises an objective function $F(\Omega)$, subject to some constraints. This section introduces the computational method used here, which is closely based on the LSTO method of Dunning and Kim [8], see also [21]. The implementation described here is a slightly simplified version of the original LSTO method: our C++ implementation is available at [29]. We give a brief and informal description of this (deterministic) optimisation algorithm, to set the scene for the stochastic method that we will describe in Sec. III. Rigorous discussions of the properties of the level-set method and of the mathematical results underlying this algorithm can be found elsewhere [17].

To define our optimisation problem, we require an objective function F , a constraint function G , and a design domain $\Omega_d \subset \mathbb{R}^d$, where d is the spatial dimensionality (In this work we take $d = 2$ although generalisation to higher dimension is possible.) The aim of our optimisation is find a structure $\Omega \subset \Omega_d$ such that $F(\Omega)$ is minimised, subject to the constraint

$$G(\Omega) \leq G^*. \quad (1)$$

Extension to multiple constraints or equality constraints of the form $G(\Omega) = G^*$ is straightforward [8, 21] but we consider a single inequality constraint here, for simplicity. The LSTO method prescribes a time-evolution for Ω that converges to a (local) optimum of F , which satisfies the constraint.

A. Evolution in continuous space and time

The structure Ω is defined in terms of a real-valued function $\phi : \Omega_d \rightarrow \mathbb{R}$. That is, define $\Omega = \{\mathbf{x} \in \Omega_d : \phi(\mathbf{x}) \geq 0\}$ as the part of the domain for which $\phi \geq 0$. Throughout this work, bold symbols such as \mathbf{x} indicate vectors in \mathbb{R}^d . The boundary of Ω is denoted by Γ and is defined as the zero level-set of ϕ , that is $\Gamma = \{\mathbf{x} \in \Omega_d : \phi(\mathbf{x}) = 0\}$.

The boundary Γ is made up from one or more closed curves, so we index the points in Γ by an internal coordinate $u > 0$, such that $\mathbf{X}(u) \in \Gamma$ is a point on the boundary of Ω . Also let $\mathbf{n}(u)$ be the (inward) normal vector to the boundary at the point $\mathbf{X}(u)$, and define a function ℓ' such that $\int_{u_1}^{u_2} d\ell(u) = \int_{u_1}^{u_2} \ell'(u) du$ is the length of the boundary between the two points $\mathbf{X}(u_1)$ and $\mathbf{X}(u_2)$.

Several level-set topology optimisation approaches [2, 6] use a steepest descent strategy for the minimisation of $F(\Omega)$. To achieve this, let the function ϕ evolve as a

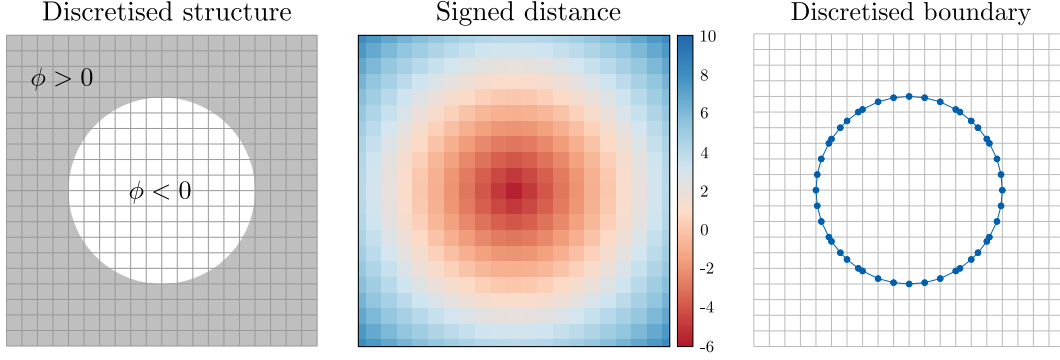


FIG. 1: The level-set domain: (Left) A two-dimensional square domain Ω_d is discretised using a uniform square grid. A function ϕ defined on the nodes of the grid indicates whether each node is inside ($\phi < 0$) or outside ($\phi > 0$) a structure Ω . In this case Ω is a square that contains a circular hole. (Middle) The function ϕ is given by the signed distance to the nearest point on the boundary of Ω . With this choice $|\nabla\phi| = 1$. (Right) A discretised representation of the boundary of Ω is obtained by defining boundary points, which are located either on nodes (if $\phi_i = 0$) or on edges between nodes (if this edge connects two nodes with opposite signs for ϕ). The resulting set of boundary points provide a piecewise linear approximation for of the boundary of Ω .

function of the time t according to an advection equation

$$\frac{\partial}{\partial t}\phi(\mathbf{x}, t) = -\mathbf{v}(\mathbf{x}, t) \cdot \nabla\phi(\mathbf{x}, t) \quad (2)$$

where \mathbf{v} is a local velocity, to be specified below. Assume that for $t = 0$ then ϕ solves the eikonal equation $|\nabla\phi| = 1$. Given that $\phi = 0$ on the boundary Γ , this is equivalent to taking $\phi(\mathbf{x})$ to be the (signed) distance of point \mathbf{x} from this boundary. The vector $\nabla\phi$ is normal to the level sets of ϕ , and one sees from (2) that the time-evolution of ϕ depends only on the normal velocity $v^n = \mathbf{v} \cdot \nabla\phi/|\nabla\phi|$. In what follows we take $\nabla v^n \cdot \nabla\phi = 0$ which ensures that if the eikonal equation is true at $t = 0$, then $|\nabla\phi| = 1$ for all times $t > 0$. Note that for points on the boundary Γ , the inward normal is $\mathbf{n}(u) = \nabla\phi(\mathbf{x}(u))/|\nabla\phi(\mathbf{x}(u))|$.

Hence, to specify the time evolution of ϕ , it is sufficient to specify v^n for all points on the boundary Γ of the structure Ω , since the condition $\nabla v^n \cdot \nabla\phi = 0$ then specifies v^n at all other points. In order that (2) corresponds to steepest descent for the objective function F , we introduce a sensitivity s^F , so that $s^F(u)$ is the sensitivity for F at the point $\mathbf{X}(u)$. To define s^F , consider a deformed structure Ω_ϵ whose boundary Γ_ϵ consists of points $\mathbf{X}_\epsilon(u) = \mathbf{X}(u) + \epsilon z(u)\mathbf{n}(u)$ where the function z sets the size of the displacement of each boundary point. Informally, the sensitivity $s^F(u)$ is the rate of change of F associated with moving the boundary point $\mathbf{X}(u)$ in the direction $\mathbf{n}(u)$. More precisely, s^F is the unique function that obeys

$$F(\Omega_\epsilon) = F(\Omega) + \epsilon \int_{\Gamma} s^F(u) z(u) d\ell(u) + O(\epsilon^2). \quad (3)$$

We assume in the following that the objective function F and the boundary Γ are sufficiently smooth that this sensitivity function exists. For a rigorous discussion of these issues, see [1].

To perform an unconstrained minimisation of $F(\Omega)$, one should prepare an initial condition in which ϕ is the signed distance from the boundary of some initial design Ω_0 . Then one should solve (2), with the normal velocity at boundary point $\mathbf{X}(u)$ being

$$v^n(\mathbf{X}(u)) = -s^F(u). \quad (4)$$

The normal velocity v^n at any point \mathbf{x} which is not on the boundary Γ should be set equal to the normal velocity at the nearest point on Γ , which ensures that $\nabla v^n \cdot \nabla\phi = 0$, as noted above. This time-evolution for ϕ encodes a time-evolution for the structure Ω : given that ϕ evolves in this way, it is easy to verify that $\partial_t F(\Omega) = -\int s^F(u)^2 d\ell(u) \leq 0$, so the objective function decreases with time.

We now consider minimisation of $F(\Omega)$ subject to a constraint, that $G(\Omega) \leq G^*$. To achieve this, a Lagrange multiplier μ^G is introduced [21], such that

$$v^n(\mathbf{X}(u)) = -s^F(u) + \mu^G s^G(u) \quad (5)$$

where s^G is the sensitivity for the constraint function G , and μ^G is chosen (independent of u) such that the system does not violate the constraint on G . The generalisation of this construction to systems with multiple constraints or to equality constraints of the form $H(\Omega) = 0$ is straightforward but we restrict here to just one constraint, for compactness of notation.

B. Discrete space and time, and boundary discretisation

For a computational implementation, these equations must be discretised in both space and time. For the spatial discretisation, the domain Ω_d is partitioned into a square grid of side $a_0 = 1$: see Fig. 1. The vertices of the grid are called nodes. Each node i has a position \mathbf{x}_i and

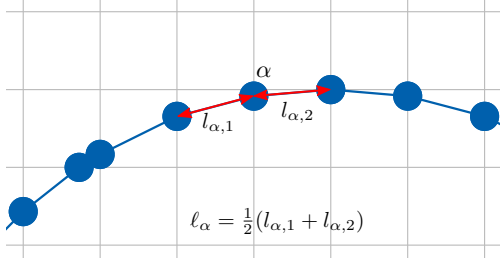


FIG. 2: Enlarged view of the discretised zero contour of the level set function. Each boundary point α is associated with two segments of the piecewise linear boundary. For each boundary point, we define an associated boundary length $\ell_\alpha = (l_{\alpha,1} + l_{\alpha,2}) / 2$.

an associated value of the level-set function $\phi_i = \phi(\mathbf{x}_i)$. The temporal discretisation is a simple first-order Euler scheme, so (2) becomes

$$\phi_i(t + \Delta t) = \phi_i(t) - v_i^n(t) \Delta t \cdot |\nabla \phi(t)|_i \quad (6)$$

where v_i^n is the normal velocity at node i and $|\nabla \phi(t)|_i$ is the modulus of $\nabla \phi$, evaluated at \mathbf{x}_i , which is equal to unity if ϕ solves the eikonal equation exactly. In practice, $\nabla \phi$ is estimated for each node using the Hamilton-Jacobi weighted essentially non-oscillatory method (HJ-WENO) described in [17].

In order to determine the normal velocities $\{v_i^n\}$, the LSTO method employs a second level of discretisation: see Fig. 2. From the (discretised) level set function ϕ , a (discrete) set of boundary points is inferred, as follows: If $\phi_i = 0$ then node i is a boundary point. Also, if two adjacent nodes have ϕ -values with opposite signs then there is a boundary point between them, which is taken to lie on the edge between the nodes, with a position determined by linear interpolation. Let the number of boundary points be n and let the position of boundary point α be \mathbf{X}_α (with $1 \leq \alpha \leq n$). The boundary points form a set of closed curves, which provide a discrete representation of the boundary Γ .

The reason for this boundary discretisation is that the LSTO method uses estimates of the sensitivities s^F, s^G on the boundary points (this is the natural choice since the sensitivity is intrinsically related to the boundary Γ). Given these sensitivities, one infers a velocity V_α^n for each boundary point α : these velocities are determined by a linearised optimisation sub-problem that is solved at each time step. From the boundary point velocities V_α^n , the normal velocities v_i^n for each node are calculated by a fast-marching method. Hence the level set variables can be propagated forward in time, according to (6). We now describe these steps in more detail.

C. Determination of boundary point velocities from an optimisation sub-problem

To determine the boundary point velocities V_α^n , we suppose that estimates of the sensitivities s^F and s^G are available for each boundary point. For boundary point \mathbf{X}_α let these estimates be s_α^F, s_α^G . (Estimation of these sensitivities depends on the problem of interest and will be discussed in Sec. IV.) Also define $l_{\alpha,1}$ and $l_{\alpha,2}$ as the distances from point α to its neighbouring boundary points; and let $\ell_\alpha = (l_{\alpha,1} + l_{\alpha,2}) / 2$ be the length of boundary associated with point α , as in Fig. 2. (It follows that $\sum_{\alpha=1}^n \ell_\alpha$ is the total boundary length.) Now suppose that each boundary point \mathbf{X}_α moves a distance $V_\alpha^n \Delta t$ in the normal direction: discretising (3) along the boundary, the change in the objective and constraint functions can be estimated as

$$\begin{aligned} \Delta F &= \sum_{\alpha=1}^n V_\alpha^n s_\alpha^F \ell_\alpha \Delta t, \\ \Delta G &= \sum_{\alpha=1}^n V_\alpha^n s_\alpha^G \ell_\alpha \Delta t. \end{aligned} \quad (7)$$

Direct optimisation of the V_α^n can then be used to optimise the change in F , given any constraints (see for example [8]).

However, optimising over all the parameters V_α^n is not convenient numerically: there is a large number of such parameters, and discretisation errors can result in rough boundaries Γ [8]. Instead, the LSTO method uses the (constrained) steepest-descent defined by (5), with a variable time step that is optimised according to the values of the sensitivities. Spatial discretisation of (5) yields

$$V_\alpha^n \Delta t = \lambda^F s_\alpha^F + \lambda^G s_\alpha^G, \quad (8)$$

where λ^F, λ^G are parameters to be determined (with $-\lambda^F$ corresponding to the time step Δt in (5) and λ^G corresponding to $\mu^G \Delta t$).

At each iteration, the LSTO method optimises the parameters λ^F, λ^G , in order to make ΔF as negative as possible. However, this procedure is subject to several constraints, which include both the optimisation constraint (1), and several additional considerations, which are determined by the physical characteristics of the topology optimisation problem. Note that the handling of these constraints in the algorithm presented here differs from the method of [8]. The method presented here has been chosen to combine simplicity and accuracy.

First, note that $\Delta F, \Delta G$ are *estimates* for the changes in F, G , which are accurate only if the boundary point displacements are not too large. To avoid very large values of the λ parameters, a constraint is applied to each boundary point displacement:

$$|V_\alpha^n \Delta t| \leq d_{\text{CFL}} \quad (9)$$

Note that consistency between (8) and (5) shows that $-\lambda^F$ is equal to the timestep Δt : the primary role of this

constraint is to impose a sufficiently small time step for the Euler discretisation (6). To determine an appropriate value for d_{CFL} , we use the Courant-Friedrichs-Levy (CFL) condition associated with (2), which requires that d_{CFL} be smaller than the grid spacing. Large d_{CFL} leads to larger time steps and hence faster optimisation, but if the resulting convergence histories are interpreted as approximate solutions to (2) then these solutions are less accurate when d_{CFL} is large. The specific value of d_{CFL} depends on the problem of interest.

To implement the CFL constraint, the optimisation domain for λ^F is $|\lambda^F| \leq \lambda_{\text{max}}^F$ with $\lambda_{\text{max}}^F = -d_{\text{CFL}}/(\max_{\alpha} |s_{\alpha}^F|)$. A similar restriction is applied to λ^G . Since these constraints are applied separately to λ^F and λ^G , the resulting solution may still violate the constraint (9): if this happens then the resulting λ parameters are rescaled by a factor $d_{\text{CFL}}/(\max_{\alpha} |V_{\alpha}^n \Delta t|)$, and the V_{α}^n are recalculated using (8), so that (9) is then satisfied.

Second, if the design at time t is Ω_t , the inequality constraint (1) for the optimisation requires $G(\Omega_t) + \Delta G \leq G^*$. If Ω_t does not satisfy the constraint ($G(\Omega_t) > G^*$) then the optimiser may not be able to find any solution for which ΔG is sufficiently negative to solve the constraint. For this reason, the optimisation is performed subject to a modified constraint $\Delta G \leq -G_0$ where $G_0 = G(\Omega_t) - G^*$ if a solution is possible, otherwise a smaller value for G_0 is chosen. (In practice, the smallest (most negative) possible value for ΔG is calculated by considering the cases $\lambda^{F,G} = \pm \lambda_{\text{max}}^{F,G}$: if the most smallest possible value ΔG^{min} is greater than $G^* - G(\Omega_t)$ then it is likely that the constraint cannot be satisfied so we take $G_0 = -c \Delta G^{\text{min}}$, where the parameter c can be adusted according to the problem of interest. Typically we take $c \approx 0.5$. Note that $\Delta G^{\text{min}} > G^* - G(\Omega_t)$ usually happens only during the initial stages of optimisation, so the way that this case is handled does not typically affect the convergence of the method to its final solution.)

Third, note that boundary points should not move outside the design domain Ω_d . To avoid this problem, let d_{α} be the signed distance of boundary point α from the boundary of Ω_d . (The sign of d_{α} is positive if the normal vector \mathbf{n}_{α} points towards the domain boundary, negative otherwise.) We estimate that point α has moved outside the domain if $V_{\alpha}^n \Delta t > d_{\alpha} > 0$ or $V_{\alpha}^n \Delta t < d_{\alpha} < 0$ (this is an approximation since the normal velocity is not perpendicular to the interface, but it is sufficient for our purposes). If this happens we replace (8) by $V_{\alpha}^n \Delta t = d_{\alpha}$.

With these ingredients in place, we finally define the full optimisation problem that is used to determine λ^F, λ^G . Combining (7,8), and accounting for the possibility that boundary points might move outside of the domain Ω_d , define

$$\Delta \hat{F}(\lambda^F, \lambda^G) = \sum_{\alpha=1}^n s_{\alpha}^F z_{\alpha} \ell_{\alpha}, \quad \Delta \hat{G}(\lambda^F, \lambda^G) = \sum_{\alpha=1}^n s_{\alpha}^G z_{\alpha} \ell_{\alpha} \quad (10)$$

where $z_{\alpha} = \lambda^F s_{\alpha}^F + \lambda^G s_{\alpha}^G$ if the boundary point re-

mains inside the domain, and $z_{\alpha} = d_{\alpha}$ otherwise. We then choose $\lambda^{F,G}$ to minimise $\Delta \hat{F}(\lambda^F, \lambda^G)$ subject to $\Delta \hat{G}(\lambda^F, \lambda^G) \leq G_0$, on the domain $|\lambda^F| < \lambda_{\text{max}}^F, |\lambda^G| < \lambda_{\text{max}}^G$. Given the $\lambda^{F,G}$ solving the optimisation problem, the time step is $\Delta t = -\lambda^F$ and the boundary point velocities are $V_{\alpha}^n = z_{\alpha}/\Delta t$.

In practice, this two-parameter optimisation sub-problem is solved using the SLSQP method from the NLOPT package [28]. The simple form of (10) means that derivatives of $\Delta \hat{F}, \Delta \hat{G}$ can be obtained analytically without the need for the finite differences used in [8]. If some sensitivities are very large, it is convenient to precondition the optimiser by defining rescaled parameters $\tilde{\lambda}^F = \lambda^F/A$ and corresponding rescaled sensitivities $\tilde{s}^F = A s^F$ for some parameter A , to ensure efficient solution of this sub-problem.

D. Level set update

Having calculated the boundary point velocities V_{α}^n , the velocities v_i^n on nodes adjacent to the boundary points are calculated by inverse-squared distance weighting, as in [9]. That is, if the edges associated with node i of the grid include m boundary points, then these points are indexed by $\beta = 1 \dots m$. Let the velocity of point β be V_{β}^n and its distance from node i be r_{β} . Then $v_i^n = \sum_{\beta=1}^m (V_{\beta}^n / r_{\beta}^2) / [\sum_{\beta=1}^m (1/r_{\beta}^2)]$. This fixes the v_i^n on a narrow strip that contains the boundary Γ . To fix the v_i^n on the other nodes requires a velocity extension procedure, for which we use a fast-marching method, as in [8].

Given the v_i^n on all nodes, the level set variables ϕ_i are updated according to (6). The whole process – inference of boundary points; sensitivity calculation; optimisation of λ^F and λ^G ; velocity extension; level set update – is repeated until the algorithm converges to an optimal structure.

Finally, note that in practice, it is convenient to do the velocity extension and the level set update only in a narrow band close to the boundary. This improves the efficiency of the method but it means that ϕ_i is given by the signed distance to the boundary only within the narrow band. To correct for this effect, all of the ϕ_i variables are periodically reinitialised to be consistent with a signed distance function. This reinitialisation uses the same fast-marching implementation used for the velocity extension.

Note that given a set of boundary points \mathbf{X}_{α} , reinitialisation of the ϕ_i followed by a recalculation of the boundary point positions does in general lead to small changes in the boundary point positions (see appendix). As the system gets close to convergence, this can have small but significant effects on the objective function, which acts as a weak source of numerical noise during optimisation. Whilst this does not manifest itself as a significant concern in deterministic optimisation, it has implications for the accuracy of the stochastic method described below.

III. STOCHASTIC LEVEL SET

Having described the deterministic optimisation algorithm of [8, 21], the next step is to introduce a stochastic component to this algorithm. Several other stochastic level-set methods have been considered recently [12, 13, 27], but differ from the approach proposed here in that the noise in this scheme is applied directly only on the boundary Γ , which ensures that the function ϕ retains its property as a signed-distance function under the stochastic time-evolution.

We first consider the optimisation problem for $F(\Omega)$ in the absence of any constraint on $G(\Omega)$. In this case the deterministic algorithm corresponds to steepest descent for F , which converges to a local optimum. By contrast, the corresponding stochastic algorithm does not converge to an optimum of F . Instead it converges to a steady state in which it explores a range of structures Ω . The algorithm is designed so that the probability p_T that it visits a structure Ω within the steady state is proportional to a Gibbs-Boltzmann factor

$$p_T(\Omega) \propto e^{-F(\Omega)/T} \quad (11)$$

where T is a noise intensity which would be identified as a temperature. For $T \rightarrow 0$ we see that the probability will concentrate close to the optimal structure Ω^* , which minimises F .

The advantage of the stochastic algorithm is that it can explore many different (non-optimal) designs. In particular, for a non-convex optimisation problem and given a long enough time t , it should explore all the local optima, not just the one closest to the starting point (which would be found by a deterministic method). Moreover, the distribution of structures that the algorithm finds is controlled via (11). This means that the steady states of the algorithm at different temperatures are related, as

$$\frac{p_{T_2}(\Omega)}{p_{T_1}(\Omega)} \propto e^{F(\Omega)(1/T_1 - 1/T_2)} \quad (12)$$

and in particular, the marginal distribution of the objective function itself satisfies

$$P_{T_2}(F) \propto e^{F(1/T_1 - 1/T_2)} P_{T_1}(F) \quad (13)$$

Such relationships between distributions form the basis of simulated annealing and parallel tempering methods [11, 18, 23], which have proven useful in exploring many non-convex optimisation problems.

Note however that these results are based on the proportionality relationship (11). More generally, we expect that the invariant measure of the stochastic process for the structures Ω is

$$dp_T(\Omega) = \frac{1}{Z(T)} e^{-F(\Omega)/T} dp^0(\Omega) \quad (14)$$

where $Z(T)$ is a normalisation constant and p^0 is a reference measure for structures (independent of T). We

do not have a precise characterisation of the measure p^0 , but as long as p^0 is independent of T then (12,13) are valid, and can be used to check consistency between our algorithm and the asserted invariant measure (14).

A. Stochastic motion of boundary points

To describe our stochastic method, it is useful to cast (6) as an equation of motion for the boundary points \mathbf{X}_α :

$$\frac{d}{dt} \mathbf{X}_\alpha(t) = V_\alpha^n(t) \mathbf{n}_\alpha(t). \quad (15)$$

where \mathbf{n}_α is a unit vector in the direction of the (inward-pointing) normal to the boundary Γ at point \mathbf{X}_α . The stochastic element of the dynamics operates directly on the boundary points. To implement this, Eq. (15) is replaced by a *stochastic differential equation* (SDE). For the deterministic problem of minimisation of F in the absence of any constraint, we write $V_\alpha^n = -s_\alpha^F$, consistent with (4). Then a natural generalisation of (15) is the SDE:

$$d\mathbf{X}_t^\alpha = -s_\alpha^F \mathbf{n}_\alpha dt + \mathbf{n}_\alpha \sqrt{2T/\ell_\alpha} \circ dW_t^\alpha \quad (16)$$

The theory of such equations is discussed (for example) in [16]. Roughly speaking, one may interpret $d\mathbf{X}_t^\alpha$ as a small increment in the boundary point position \mathbf{X}^α , associated with a small time increment dt . The increment $d\mathbf{X}_t^\alpha$ consists of a deterministic part $-s_\alpha^F \mathbf{n}_\alpha dt$ that is proportional to dt , and a random part $\mathbf{n}_\alpha \sqrt{2T/\ell_\alpha} \circ dW_t^\alpha$. The \circ indicates that the SDE is written in the Stratonovich convention: the implications of this will be discussed below. The increment dW_t^α is a standard white noise (or Wiener process) associated with boundary point α , which is independent of the noises on all other boundary points.

Inspection of (16) shows that the noise term acts in a direction perpendicular to the boundary (as might be expected); it is proportional to \sqrt{T} which sets the noise strength. The factor of $\sqrt{1/\ell_\alpha}$ might not be expected a priori – it is necessary because consistency of the stochastic level-set method requires that the noise intensity is equal at all points on the boundary Γ , but the boundary points are not equally spaced along Γ (recall Fig. 2). The idea is that a given boundary Γ has many possible discretisations in terms of boundary points, but the resulting stochastic evolution should be independent of this discretisation. Mathematically, this idea can be encapsulated as a reparameterisation invariance of the equations of motion, as we now discuss.

B. Evolution of continuous boundaries, and reparameterisation invariance

It is useful to consider a process by which a continuous curve evolves in time, and to interpret (16) as a discretised approximation of this process. For the continuous

curve we write a generalisation of (4), as

$$d\mathbf{X}_t(u) = -s^F(u)\mathbf{n}(u)dt + \mathbf{n}(u)\sqrt{2T/\ell'(u)} \circ dW_t(u) \quad (17)$$

where $W_t(u)$ a random Brownian noise with equal intensity at each point u . (Technically, the quadratic variation of this process satisfies $d\langle \int_{u_1}^{u_2} W_t(u)du \rangle = (u_2 - u_1)$.) To derive (16) as a discretised version of (17), we identify \mathbf{X}^α as the centre of mass of a small segment of the boundary: $\mathbf{X}_t^\alpha = (1/\ell_\alpha) \int_{u_1}^{u_2} \mathbf{X}_t(u) d\ell(u)$. Assuming (see below) that the segment is small enough that $\mathbf{n}(u)$ and $\ell'(u)$ and $s^F(u)$ can be replaced by their mean values within the integral, this yields (16).

Within this continuous setting, the idea that the evolution of the curve should be independent of its discretisation in terms of boundary points has a mathematical statement in terms of reparameterisation of the coordinate u . Consider a closed curve described by the internal co-ordinate $u \in [0, 1]$, so $\mathbf{X} : [0, 1] \rightarrow \Omega_d$ is a smooth function with $\mathbf{X}(u)$ a point on the curve. Now consider a continuous monotonically-increasing function $f : [0, 1] \rightarrow [0, 1]$ with $f(0) = 0$ and $f(1) = 1$. Define $\tilde{\mathbf{X}} : [0, 1] \rightarrow \Omega_d$ by $\tilde{\mathbf{X}}(u) = \mathbf{X}(f(u))$. It should be clear that \mathbf{X} and $\tilde{\mathbf{X}}$ are different representations of the same curve.

In order that the evolution of a curve Γ does not depend on its parameterisation in terms of boundary points, we require that (17) evolves the functions \mathbf{X} and $\tilde{\mathbf{X}}$ in the same way. This may be verified by direct substitution, as long as the noise prefactor $\sqrt{1/\ell'(u)}$ is included. This is the reason for including the factor in $\sqrt{1/\ell_\alpha}$ that multiplies the noise in (16).

We have assumed throughout that the boundary Γ is smooth enough and the normal vector $\mathbf{n}(u)$ can be defined at every point $\mathbf{X}(u)$. Even for deterministic optimisation, this assumption requires some care, if the optimal shape has a boundary that includes kinks. For stochastic optimisation, there is an additional factor, which is that the noise tends to roughen the curve and the normal vector $\mathbf{n}(u)$ may not be defined. In our numerical scheme, the boundary is discretised and a WENO estimate of $\nabla\phi$ can always be used to define a local normal, so potential problems with rough boundaries do not appear. For a detailed mathematical analysis, some regularisation is required to ensure existence and uniqueness of the SDE (17), but we defer these issues to a later publication.

C. Invariant measure

Our assertion is that the invariant measure for the stochastic process described in Sec. IIIB is proportional to (11). We do not have a rigorous proof of this claim, although we will demonstrate below that our numerical method is consistent with (13). As a plausibility argument for our scheme, we note that if $\mathbf{n}_\alpha s_\alpha^F$ in (16) is equal

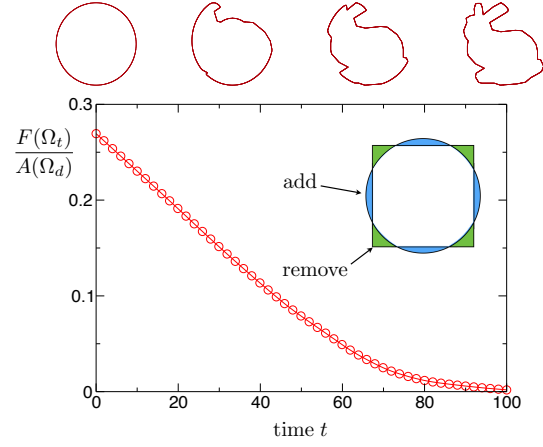


FIG. 3: Shape matching by the level-set method (without any stochastic component). The target shape is a discretised representation of the Stanford bunny (top right); the initial shape is a circle (top left). The main figure shows a time series of the (normalised) objective function $F(\Omega_t)/A(\Omega_d)$ with F given by (24) and $A(\Omega_d)$ the total area of the design domain. Representative shapes obtained during this convergence procedure are also shown (above). The inset illustrates the definition of the objective and sensitivity functions, for the simpler case of a circular target shape and a square initial shape Ω . Blue shading indicates regions where $\phi_i^{\text{target}} > \phi_i$ and $s^F = 1$, so inward motion of the boundary increases F . Similarly green regions have $\phi_i^{\text{target}} < \phi_i$ and $s^F = -1$. The normalised objective function $F(\Omega)/A(\Omega_d)$ is given by the sum of the blue and green areas, as a fraction of the total area of the design domain Ω_d .

to $(\partial F/\partial \mathbf{X}_\alpha)\ell_\alpha^{-1}$ then (16) has the form

$$d\mathbf{X}_t^\alpha = (\partial F/\partial \mathbf{X}_\alpha)\sigma^2 dt + \sigma\sqrt{2T} \circ dW_t^\alpha \quad (18)$$

with $\sigma = 1/\sqrt{\ell_\alpha}$. It is well-known [16] that the invariant measure for this equation is of the form (11). Moreover, spatially discretising (3) and identifying $\epsilon z(u)$ as the boundary point displacement $\mathbf{n}_\alpha \cdot \delta \mathbf{X}_\alpha$ yields

$$F(\Omega_\epsilon) - F(\Omega) = \sum_\alpha s_\alpha^F \ell_\alpha \mathbf{n}_\alpha \cdot \delta \mathbf{X}_\alpha + O(\delta \mathbf{X}^2) \quad (19)$$

indicating that $\ell_\alpha \mathbf{n}_\alpha s_\alpha^F$ does indeed correspond to the derivative $(\partial F/\partial \mathbf{X}_\alpha)$, as required. Refining this argument into a more rigorous proof is an interesting direction for future work.

In the following, we show how the process (16) can be implemented within the level-set method described in Sec. II. We assume that (12) holds theoretically and we investigate the extent to which it holds numerically.

D. Stochastic dynamics with a finite time step

To implement the stochastic evolution (16) within the level-set method of Sec. II, we require a generalisation of

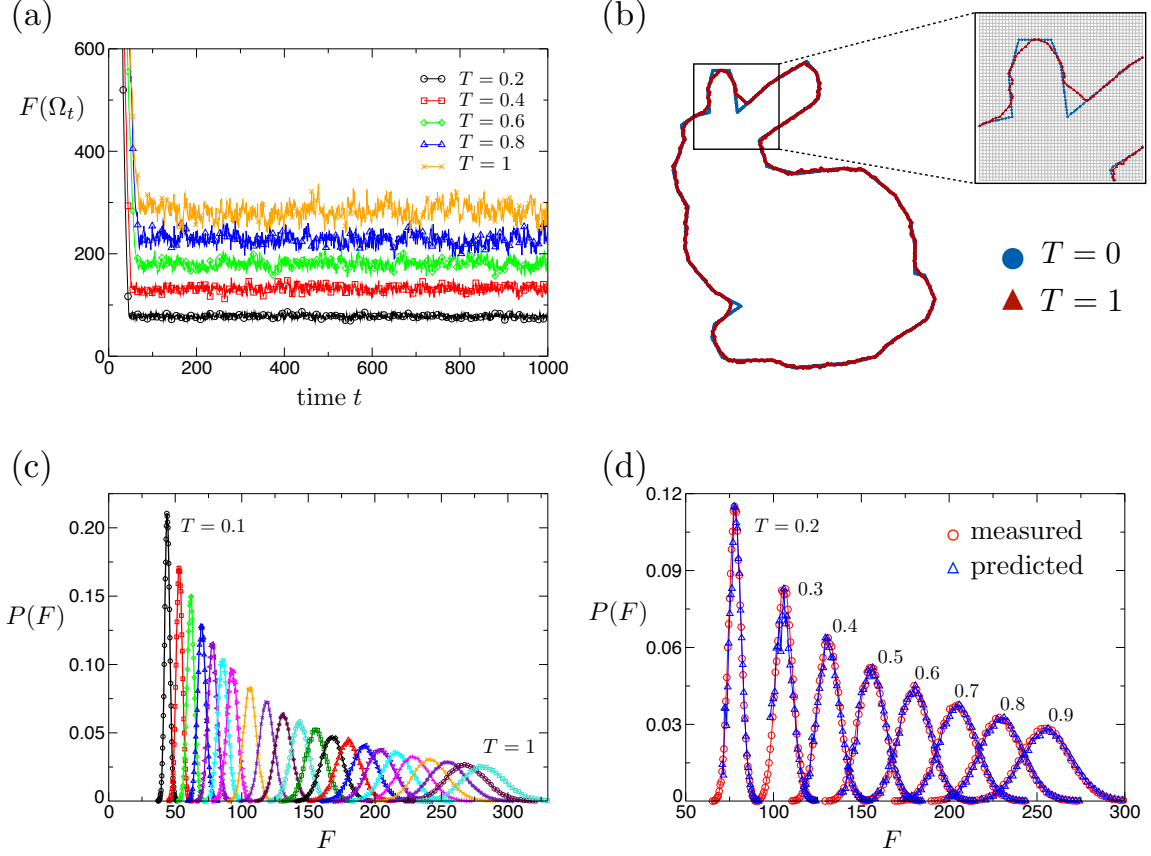


FIG. 4: Shape matching with finite noise strength T : (a) Time series of the area mismatch (24) for several noise strengths (temperatures). Raising temperature leads to increasingly sub-optimal designs and larger fluctuations. (b) A representative shape (red) from the steady state at temperature $T = 1$, compared with the target shape (red). Mismatch with the target structure is most pronounced in regions of high curvature. (c) Histograms (normalised as probability densities) of the objective function, obtained from long time series in the steady state of the system. Data are shown for seven temperatures equally spaced between $T = 0.1$ and $T = 0.25$, with the remaining temperatures equally spaced between $T = 0.25$ and $T = 1.0$. (d) The measured probability densities agree well with the predictions of Eq. (13): we show measured $p_T(F)$ for various temperatures, and the corresponding predictions using Eq. (13) with $T_1 = T$ and $T_2 = T + \Delta T$. The temperature increment ΔT is such that the distributions at T_1, T_2 are adjacent in (c).

(6). To achieve this, integrate (16) over a small time interval Δt and identify the average boundary point velocity in the normal direction as $\bar{V}_\alpha^n(t) = (\Delta t)^{-1} \int_{s=t}^{t+\Delta t} \mathbf{n}(s) \cdot d\mathbf{X}_s^\alpha$. Hence, keeping terms up to first-order in Δt :

$$\bar{V}_\alpha^n(t)\Delta t = -s_\alpha^F(t)\Delta t + \sqrt{\frac{2T\Delta t}{\ell_\alpha(t + \Delta t/2)}}\xi \quad (20)$$

where ξ is a standard normal-distributed random number. The first term on the right hand side is the standard deterministic increment corresponding to (15). The second term is a random increment (with zero mean). Note however that the length ℓ_α in this second term is evaluated at time $t + \Delta t/2$, which corresponds to the midpoint of the time interval (this is the meaning of the Stratonovich product that was denoted by \circ in (16)). To arrive at an equation that can be used in a (first-order) numerical scheme, it is necessary to calculate $\bar{V}_\alpha^n(t)$ in

terms of quantities that are evaluated only at time t . This can be achieved by using Ito's formula [16], which yields

$$\bar{V}_\alpha^n(t)\Delta t = -s_\alpha^F(t)\Delta t + \sqrt{\frac{2T\Delta t}{\ell_\alpha(t)}}\xi - \frac{T\kappa_\alpha}{2\ell_\alpha}\Delta t. \quad (21)$$

where we again keep terms up to first-order in Δt , and κ_α is the (signed) curvature of the boundary at point α (see Appendix).

The derivation of (21) from (16) is straightforward within the framework of stochastic calculus. We omit technical details and provide a short argument to justify it: for a boundary point increment $d\mathbf{X}_\alpha$, the change in the length of the boundary segment is $d\ell_\alpha = \mathbf{n}_\alpha \cdot (\ell_\alpha \kappa_\alpha \circ d\mathbf{X}_\alpha)$. (Roughly speaking this corresponds to the chain rule, with $d\ell_\alpha/d\mathbf{X}_\alpha = \mathbf{n}_\alpha \ell_\alpha \kappa_\alpha$, as discussed in the Appendix.) Ito's formula [16] states that for an

SDE of the form $dx_t = f(x_t)dt + \sigma(x_t) \circ dW_t$, the appropriate first-order discretisation is $\Delta x = f(x_t)\Delta t + \sigma(x_t)\xi\sqrt{\Delta t} + \frac{1}{2}\sigma'(x_t)\sigma(x_t)\Delta t$. In this case we have from (16) that $\sigma = n_\alpha\sqrt{2T/\ell_\alpha}$: the analogue of σ' is $(d\sigma/d\ell_\alpha) \cdot (d\ell_\alpha/d\mathbf{X}_\alpha) = (-\kappa_\alpha/2)\sqrt{2T/\ell_\alpha}$, which yields (21).

E. Implementation in level-set method, and incorporation of constraints

The stochastic velocity (21) is straightforwardly incorporated into the deterministic level set method described in Sec. II. This yields our stochastic level set optimisation method. The only modification of the deterministic algorithm is that noise-dependent terms are added to (8). We consider the case of optimisation subject to the constraint (1).

In the stochastic method, the parameters λ^F, λ^G in (8) are calculated exactly as in the deterministic method. The simplest approach is then to identify $\lambda^F \rightarrow -\Delta t$ and add the stochastic terms in (21) to the boundary increments. However, since the first stochastic term is proportional to $\sqrt{\Delta t}$, this can lead to large boundary point displacements that violate the CFL condition. Hence, given a deterministic time step $\Delta t = -\lambda^F$, we calculate a typical stochastic increment $\Delta x_{\text{typ}} = \sqrt{2T\Delta t}$. If $\Delta x_{\text{typ}} > d_{\text{CFL}}/2$, the deterministic parameters λ^F, λ^G are rescaled by a factor $d_{\text{CFL}}/2\Delta x_{\text{typ}}$. That is

$$\begin{aligned}\lambda_{\text{stoch}}^F &= \lambda^F \min(1, d_{\text{CFL}}/2\Delta x_{\text{typ}}) \\ \lambda_{\text{stoch}}^G &= \lambda^G \min(1, d_{\text{CFL}}/2\Delta x_{\text{typ}}).\end{aligned}\quad (22)$$

This construction amounts to fixing a maximal time step that ensures that the CFL condition is obeyed even in the presence of the noise, as long as none of the ℓ_α parameters are too small (compared to unity). (We also tested a method where the Δx_{typ} is adjusted to account for the possibilities that some ℓ_α s are very small, but this led to a less efficient method and had very little effect on the results.) The time step is $-\lambda_{\text{stoch}}^F$ and the boundary point increment is finally

$$\bar{V}_\alpha^n \Delta t = s_\alpha^F \lambda_{\text{stoch}}^F + \sqrt{\frac{2T|\lambda_{\text{stoch}}^F|}{\ell_\alpha}} \xi + \frac{T\kappa_\alpha}{2\ell_\alpha} \lambda_{\text{stoch}}^F + \lambda_{\text{stoch}}^G s_\alpha^G \quad (23)$$

This equation replaces (8) within our stochastic level set method. Once the boundary point velocities \bar{V}_α^n have been calculated in this way, the rest of the method follows exactly the deterministic case: the velocities on the nodes are calculated by interpolation and fast-marching, and the level set variables are updated using (6).

IV. RESULTS

In the following we consider several examples of the

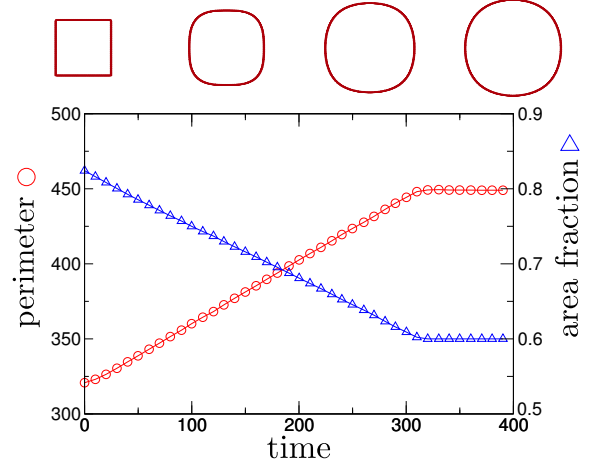


FIG. 5: Time series for deterministic minimisation of shape perimeter (circles, left axis), subject to a constraint that the area fraction *outside* Ω must be at most 60% of the total area (triangles, right axis). The top panel shows the time-dependence of Ω , starting from a square initial shape, which violates the constraint. The shape Ω increases in size (to satisfy the constraint) and evolves towards a circle (to minimise the perimeter). The domain Ω_d is a square grid of size 200×200 .

stochastic level set method, with increasing levels of complexity

A. Shape matching

We first consider a simple geometric optimisation problem. The idea, illustrated in Fig. 3 is that the domain Ω should match a predefined reference shape Ω_{target} , in this case a discretised two-dimensional representation of the Stanford bunny [24]. The objective function is

$$F(\Omega) = \sum_i |A_i^{\text{target}} - A_i(\Omega)| \quad (24)$$

where the sum runs over the cells of the grid, $A_i(\Omega)$ is the overlap area between the grid cell i and Ω , and similarly A_i^{target} is the overlap area between element i and Ω_{target} .

To calculate sensitivities, note that if A is the area of Ω then $s_\alpha^A = -1$ for all boundary points α . For the objective function (24), this requires $s_\alpha^F = 1$ if boundary point α is inside Ω_{target} and $s_\alpha^F = -1$ otherwise. To estimate this sensitivity based on local information, we calculate for each node i its signed distance ϕ_i^{target} from the boundary of the target shape. For each boundary point α , we estimate its signed distance from the boundary of the target as $\Phi_\alpha^{\text{target}}$, which is interpolated from the ϕ_i^{target} values on the four nearest nodes. We also interpolate a value $\Phi_\alpha^{\text{trial}}$ by applying the same method to the nearest values of ϕ_i : since the boundary point is on the zero contour of ϕ we expect $|\Phi_\alpha^{\text{trial}}| \ll 1$ for all α .

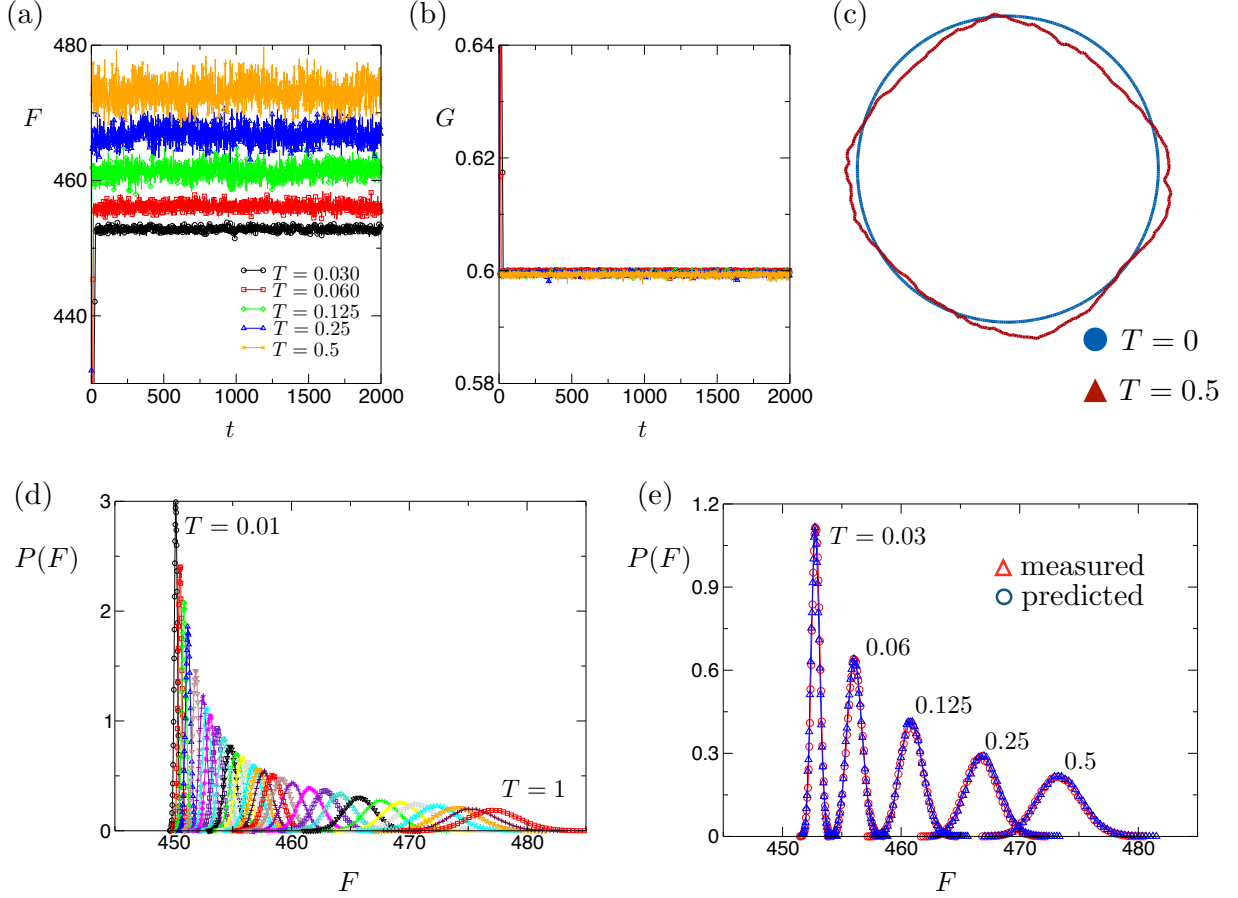


FIG. 6: Stochastic optimisation of perimeter with a constraint on the enclosed area. (a,b) Time series for the objective function F (perimeter) and the constraint function G (area). The legend in (a) applies to both panels. (c) The zero-temperature optimum (a circle), compared with a representative sample from a stochastic calculation at $T = 0.5$. (d) Histograms of the objective function, taken from the steady state of the stochastic process, at various temperatures. (e) Measured probability densities, compared with the predictions of Eq. (13). The predictions use Eq. (13) with $T_1 = T$ and $T_2 = T + \Delta T$, as in Fig. 4. The agreement is good.

The sensitivity is then estimated as

$$s_{\alpha}^F = \text{sign}(\Phi_{\alpha}^{\text{target}} - \Phi_{\alpha}^{\text{trial}}) \quad (25)$$

Since $\Phi_{\alpha}^{\text{trial}}$ is small in magnitude, its inclusion in (25) has a small effect when the current and target shapes are different from each other, but ensures that the deterministic method converges (exactly) to $\Omega = \Omega^{\text{target}}$.

Based on these sensitivities, Fig. 3 shows the deterministic optimisation algorithm in operation. An initially circular structure evolves in time until it matches exactly the target shape. The CFL constraint is $d_{\text{CFL}} = 0.1$ and the domain is partitioned into a grid with 200×200 elements, with each element having size 1. The initial shape is a circle of radius 50.

Fig. 4 shows results for the stochastic algorithm, applied to the same problem. Panel (a) shows that the objective function does not converge to its optimal value ($F = 0$). Instead the system converges to a steady state in which F fluctuates around a non-zero mean value. As

the noise strength increases, both the mean value of F and its fluctuations increase, as the algorithm explores a range of non-optimal designs. Panel (b) shows a representative non-optimal shape. Panel (c) shows distributions (histograms) of the objective function, for different values of the noise. Finally in panel (d) we test the prediction of (13), that distributions of the objective function at different noise strengths can be related to each other. That is, for various temperatures T , the distribution $P(F)$ can be predicted based on its distribution at a different temperature $T + \Delta T$. (This prediction is possible only for small ΔT since otherwise the exponential factor in (13) leads to large statistical errors.)

The results shown in Fig. 4(d) represent strong evidence that the system has converged to a steady state in which the predictions of (13) are valid. The stochastic algorithm was designed in order to obtain a steady state that satisfies (11), and the numerical agreement with (13) is a stringent test of this criterion, which indicates that

the theoretical analysis and numerical implementation presented here are self-consistent.

Moreover, the nature of the non-optimal shapes shown in Fig. 4(b) reveal some information about the null measure p_0 in (14): of the shapes Ω for which $F(\Omega)$ is not equal to its optimal value, the method seems to sample preferentially those shapes with low curvature. In this sense the numerical noise seems to act to smooth the boundary Γ . A detailed analysis of this effect will require a deeper understanding of the stochastic dynamics of the boundary, which is an interesting direction for future work.

B. Perimeter minimisation with area constraint

As a simple optimisation problem that includes a constraint, we next consider minimisation of the perimeter of Ω , subject to a constraint on its total area. Given that $F(\Omega)$ is the perimeter of Ω , the sensitivity s^F is given by the (signed) Euclidean curvature κ of the boundary Γ (see Appendix). The constraint function G is the difference between the total domain size and the area of Ω : that is, $G(\Omega) = A(\Omega_d) - A(\Omega)$. (This ensures that the *lower* bound on the area of Ω can be cast as a constraint in the form (1).) The associated sensitivity is $s^G = 1$.

Estimation of the curvature κ is non-trivial for the discretised boundaries considered here [4]. Following numerical tests of several different methods, we use a scheme that combines accuracy and simplicity, by calculating curvatures using an explicit finite-difference sensitivity calculation. Details of this procedure, and its associated errors are discussed in the Appendix.

Results for deterministic optimisation are shown in Fig. 5. The grid is of size 200×200 . An initially square structure Ω evolves into a circle whose area satisfies the constraint. In this case $G \leq 0.6A(\Omega_d)$, so the area of Ω must be at least 40% of the domain Ω_d .

Fig. 6 shows results for the stochastic method, which are comparable with Fig. 4. Fig. 6(b) shows that the stochastic algorithm still satisfies the constraint (up to some numerical uncertainties). Fig. 6(e) shows that the results are again consistent with the algorithm sampling a distribution of shapes consistent with (11,14).

However, the example shape shown in Fig. 6(c) for $T = 0.5$ reveals that this numerical method is affected by discretisation from the underlying grid. In particular, the shape Ω shown in that figure is not circular, but is elongated along the lattice axes, forming a kind of diamond shape. We find that the shapes Ω found for $T > 0$ consistently have this property (data not shown). We have investigated the reason for this effect, which we attribute to uncertainties in our numerical estimates of the sensitivity parameters s_α , particularly the sensitivity of the perimeter, which is the curvature. These numerical issues are discussed in the Appendix. In terms of the general method, our conclusion is that the stochastic level-set method relies on accurate sensitivity estimates,

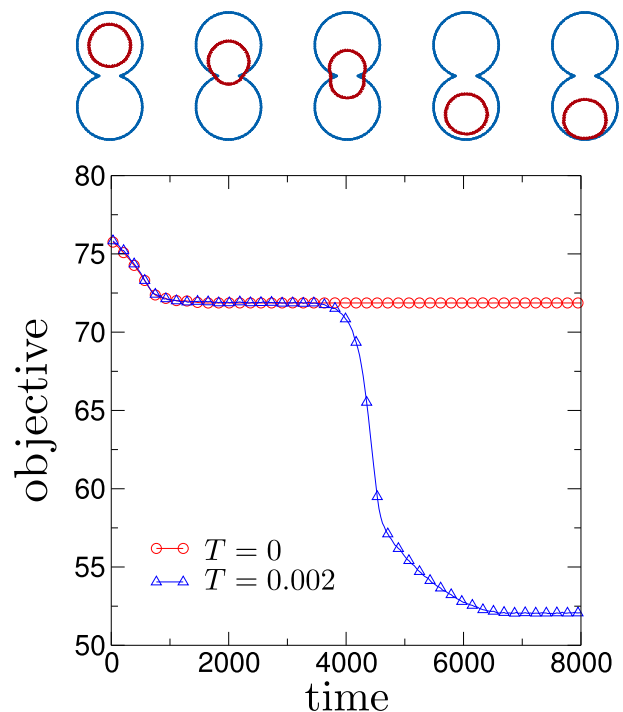


FIG. 7: Non-convex optimisation problem. (Top row) snapshots from an optimisation trajectory in which the perimeter of the red shape Ω and its height within the design domain are both being minimised, subject to a constraint on the mismatch of the red shape with the blue “dumbbell”. The global optimum is shown in the rightmost panel while the second panel illustrates an additional local optimum. (Main panel) Time series of the objective function for deterministic optimisation ($T = 0$) and stochastic optimisation ($T = 0.002$). Only in the stochastic case does the system escape from the local optimum and converge to the global one. The mesh size is 100×100 , the dumbbell consists of two circles of radius 20 whose centres are separated by $B = 38$. The parameter $\gamma = 0.65$.

which may be difficult to obtain in practice. However, we believe that the method itself is valid.

C. Non-convex shape matching problem with perimeter constraint

So far, we have focussed on very simple optimisation problems, as proof of principle for the method. However, a central motivation for the development of a stochastic method is that the noise forces can allow the system to visit multiple locally-optimal designs in a non-convex optimisation problem. As a simple example of such a problem, we consider the problem shown in Fig. 7.

The constraint function in this case is related to the overlap between the shape Ω and the dumbbell (or hour-glass) target shape shown in Fig. 7. By analogy with

(24), we define

$$G = \sum_i |A_i^{\text{target}} - A_i(\Omega)| \quad (26)$$

where the sum runs over all cells of the underlying grid. The constraint is that $G \leq 0.2A(\Omega_d)$: that is, the mismatch between the shape Ω and the target (dumbbell) shape can be at most 20% of the total design domain. In practical terms, this means that when Ω is contained within the dumbbell, the area of the dumbbell that is outside Ω must be at most $0.2A(\Omega_d)$. The dumbbell is given by the union of two circles of radius 20. In Cartesian coordinates, the size of the design domain is $L_x \times L_y$ and the centres of the circles are at $(L_x/2, (L_y \pm B)/2)$ with $B = 38$.

The objective function in this case is a weighted perimeter $F(\Omega) = M(\Omega)$ with

$$M(\Omega) = \int_{\Gamma} m(\mathbf{X}(u)) d\ell(u) \quad (27)$$

If $m(\mathbf{X}) = 1$ for all \mathbf{X} then M is simply the perimeter of Ω . The idea is that $m(\mathbf{X})$ is the weight (per unit length) of a boundary segment at position \mathbf{X} . We write $\mathbf{X} = (X, Y)$ in Cartesian coordinates and $m(\mathbf{X})$ depends only on Y . It is given by $m = 1$ for positions above the centre of the upper lobe of the dumbbell, and $m = \gamma$ for positions below the centre of the lower lobe. In between, m varies linearly with Y , so that

$$m(X, Y) = \begin{cases} 1, & Y > (L_y + B)/2 \\ \gamma, & Y < (L_y - B)/2 \\ \frac{1+\gamma}{2} + \frac{1-\gamma}{2}(Y - L_y/2), & \text{otherwise.} \end{cases} \quad (28)$$

The parameter $\gamma = 0.65$. Physically, the objective function is small when the boundary Γ is located in the lower lobe of the dumbbell, and larger when it is in the upper lobe. The optimal design is a circle located inside the dumbbell, below the centre of the lower lobe.

The sensitivity s^G for the constraint function G was described already in Sec. IV A. The sensitivity for M is $s_{\alpha}^M = \kappa_{\alpha} m(\mathbf{X}_{\alpha}) + n_{\alpha}^y \frac{\partial m(\mathbf{X}_{\alpha})}{\partial Y_{\alpha}}$, where κ_{α} is the signed curvature and n_{α}^y is the y component of the normal vector \mathbf{n}_{α} .

Fig. 7 shows results for both deterministic and stochastic optimisation for this problem. The deterministic algorithm reveals that this is indeed a non-convex optimisation problem: that is, there are two local minima of the objective function. Starting with a circular shape located in the upper lobe of the dumbbell, the deterministic algorithm converges quickly to a local optimum in which the shape is located close to the neck (2nd image in top row of Fig. 7). The global optimum has the shape Ω inside the lower lobe, but convergence to this shape is frustrated because the near-circular locally-optimal shape cannot fit through the neck of the dumbbell. In order to pass through the neck, while still obeying the constraint, the area inside the dumbbell must expand, to compensate the

extra area outside. However, this leads to an increase in the objective function F , which is not possible within the deterministic algorithm.

However, on adding a weak stochastic element, one sees (for these parameters) that the system escapes the local optimum and converges to a steady state where it samples shapes that are close to the global optimum. A final deterministic optimisation could be used to locate the true optimum, if required. This example shows the potential usefulness of the stochastic level-set optimisation method, although this is obviously a very simple model at this stage.

D. Compliance minimisation

As a final example, we show how this method can be applied to problems inspired by engineering applications of shape optimisation (this is distinct from topology optimisation in that no holes are created during optimisation). As a proof of principle, we study a well-known benchmark problem of a two-dimensional cantilevered beam [15], as shown in Fig. 8.

In this problem, the grid used within the level-set method plays a second role as a finite-element mesh for the structure. The objective function F is the strain energy of the structure. To calculate this, the structure is clamped at the left boundary ($L_x = 0$) and a unit load is applied in the y -direction at position $(L_x, L_y/2)$. The Young's modulus of the material inside Ω is $Y = 100$, the Poisson's ratio is 0.3. The space outside Ω is occupied by a weak material whose density is 10^{-3} . The strain energy is minimised subject to the constraint that the total area of Ω is at least 0.5 of the domain Ω_d . The sensitivity for the compliance is calculated by the method introduced in [8] and the sensitivity for the area constraint is $s^G = -1$, as discussed in Sec. IV A. The mesh size is $L_x \times L_y$ with $(L_x, L_y) = (40, 20)$. This relatively coarse mesh is used for computational convenience – future work will exploit more efficient sensitivity calculations which will allow access to finer grids, but this is beyond the scope of this article.

Once the sensitivities are known, the method proceeds exactly as in the simple examples considered in previous sections. The results in Fig. 8 are qualitatively consistent with Fig. 6, and show that the stochastic level set method can be applied in such contexts. The initial condition for the optimisation is a completely full grid $\Omega = \Omega_d$. One sees from Figs. 8(a,b) that the optimiser first reduces the area of Ω in order to satisfy the constraint: during this part of the algorithm the strain energy increases. Then, once the structure satisfies the constraint, its shape is optimised to reduce the strain energy. Fig. 8(c) shows the optimal design that was found by deterministic optimisation, and a design found by the stochastic method. Fig. 8(d) shows that increasing the noise increases the typical values of the objective function F , and the variance of this quantity also increases. However, Fig. 8(e)

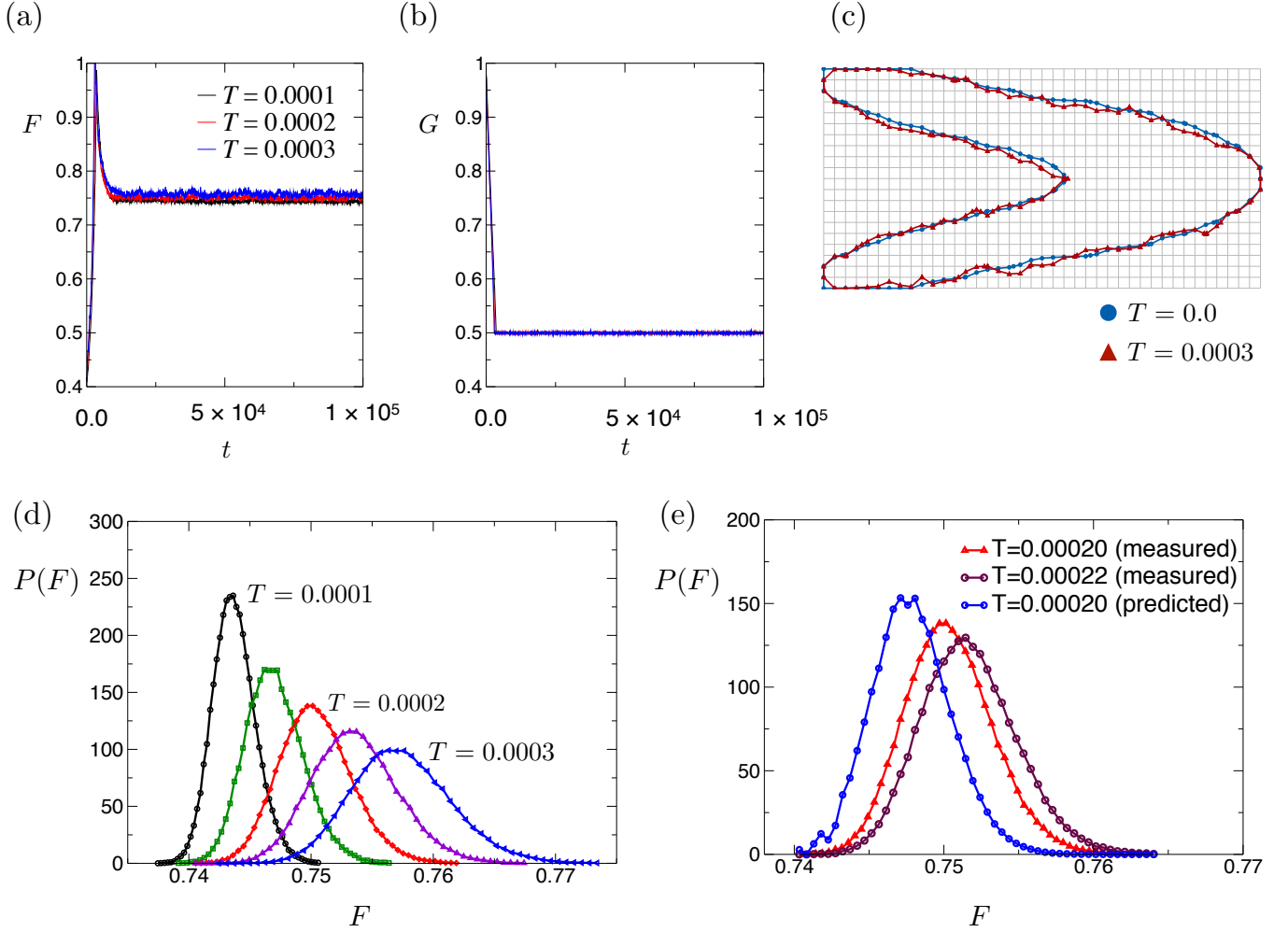


FIG. 8: Stochastic optimisation of compliance. (a,b) Time series for the objective function (strain energy) and constraint function (material area) for the model problem described in the text. (c) The (local) minimum found by deterministic optimisation ($T = 0$) and a representative structure at $T = 0.0003$. (d) Histograms for the objective function for different noise strengths T . (e) Test of the reweighting formula (13). The results are not consistent with (13): we show measured distributions at $T_1 = 0.00020$ and $T_2 = 0.00022$. The distribution at T_2 is used together in (13) to arrive at a prediction for the distribution at T_1 , but this prediction is not accurate. We attribute this effect to numerical errors associated with discretisation and sensitivity estimation, as discussed in the main text

shows that the results are not quantitatively consistent with (13). There are several potential sources of numerical error in this algorithm, of which the largest is expected to be the numerical uncertainties in calculations of sensitivities, due to the finite-element approximation. We attribute the deviations from the predictions of (13) to these numerical uncertainties.

To understand this effect in more detail, it is useful to notice that (13) implies quite generally that $\text{Var}(F) = T^2 \frac{d}{dT} \langle F \rangle$, where $\langle F \rangle$ is the mean strain energy at temperature T , and $\text{Var}(F) = \langle F^2 \rangle - \langle F \rangle^2$ is its variance. This is an example of a fluctuation-dissipation theorem (FDT) [5]. For the systems considered in Figs. 4,6, we find that this relation holds (otherwise the predictions based on the reweighting formula (13) would fail).

For this strain energy problem, we find that $\text{Var}(F) \approx 3T^2 \frac{d}{dT} \langle F \rangle$, indicating that the variance of the objective is around three times larger than the prediction given by the FDT. Our hypothesis is that these extra fluctuations in F come from numerical errors associated with discretisation and with estimates of the sensitivity, but this remains an area for future study.

Finally we note that the globally-optimal structures for this (discretised) compliance problem are not simply-connected like the shapes considered in Fig. 8(c): the optimal structures include holes [7]. The stochastic method described here does not include an explicit prescription for the creation of holes during optimisation, even if these holes would reduce the objective function. For this reason, we believe that the invariant measure of the method

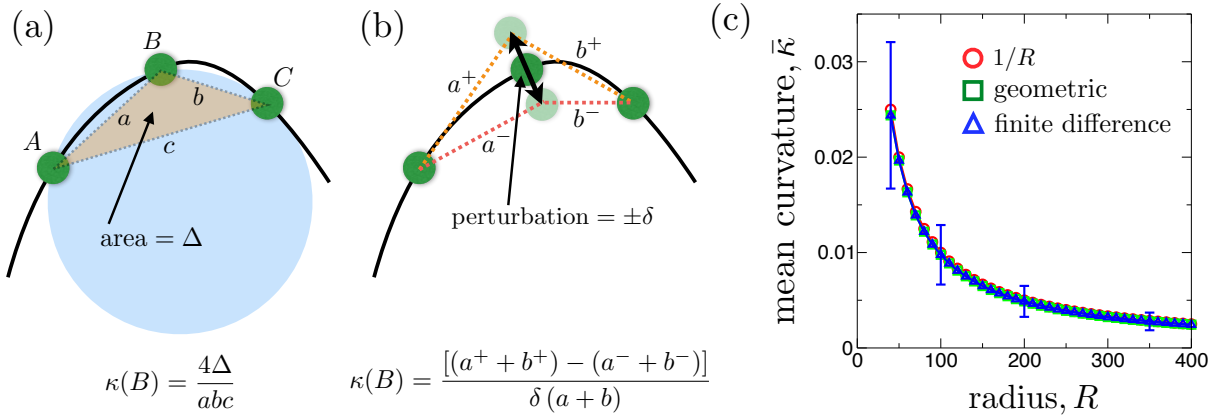


FIG. 9: Approximations for the local Euclidean curvature around a boundary point. (a) A simple geometric approximation can be obtained by matching a circle to any three sequential boundary points. The three points are assumed to lie on the perimeter of the circle, with the curvature defined as the inverse of the circle’s radius [4]. (b) The curvature can also be calculated by performing an explicit finite-difference sensitivity calculation. A boundary point is displaced in the outward and inwards directions along its normal vector and a sensitivity is defined as the rate of change of the discretised perimeter per unit length, i.e. using a central finite-difference. In the limit of $\delta \rightarrow 0$ this defines a local perimeter sensitivity for the boundary point. (c) Euclidean curvature for circles of increasing radius R . The level-set is initialised as an exact signed-distance from a circle and the curvature is measured at each boundary point, using the geometric method from (a) and the finite-difference method from (b). The mean curvature is obtained by averaging over the boundary points and is compared with the exact result $1/R$ (red circles). For the finite-difference method we also show selected error bars whose size corresponds to the standard deviation of the measured curvatures, as discussed in the text. A value of $\delta = 10^{-4}$ was used for the central difference calculation.

described here is of the form (14), with $p^0(\Omega) = 0$ if Ω is not simply connected. Generalisation of the method to include shapes with holes is an important area for future work.

V. CONCLUSIONS

We have introduced a stochastic level-set shape optimisation method, which is based on the deterministic (steepest-descent) method of [8, 21]. The stochastic element of the algorithm acts on the boundary of the shape Ω , and the method converges to a steady state in which it explores a range of shapes, according to a probability distribution (14). The method is novel – we are not able to prove rigorously that it converges to (14) but we have verified that this convergence does hold in two simple problems: shape matching (Fig. 4) and perimeter minimisation at fixed area (Fig. 6). A deeper mathematical analysis of the method would be in the scope of future work.

The motivation for introducing the stochastic method is to enable optimisation in non-convex problems, in which deterministic methods converge to local minima but are not able to search the whole parameter space in order to find the global optimum. To demonstrate the idea, we have shown results in Fig. 7 for a simple non-convex problem, in which the stochastic algorithm outperforms the deterministic one and converges to the global optimum. For complex problems with multiple optima, an important feature of the method is that the

underlying Boltzmann distribution in (14) allows it to be combined with methods such as parallel tempering, which are known to be effective in highly non-convex problems [11, 18, 23]. Finally, we considered a model engineering problem (Fig. 8), for which we demonstrated that optimisation can be performed even for problems with complicated objective functions. The inherent numerical errors which are usually insignificant in deterministic optimisation are seen to be significant in the proposed stochastic optimisation and further work is needed to investigate this.

Compared with other stochastic optimisation methods that do not use gradient (or sensitivity) information [14, 25, 26], the scheme presented here has two strengths. First, as the noise strength is reduced, it recovers to the standard deterministic method, so it is guaranteed to perform no worse than that method (this property is not guaranteed in many stochastic algorithms). Second, the fact that the invariant measure (or at least its T -dependence) is known to be (14) allows integration with parallel tempering and other methods from statistical physics [11, 18, 23], which have proven extremely valuable in that context.

Acknowledgements

We thank Li-Tien Cheng for helpful discussions. This article is based on work supported by the Air Force Office of Scientific Research, Air Force Material Command, USAF under Award No. FA9550-15-1-0316.

Appendix A: Sensitivity for the curve perimeter, local curvature and level-set reinitialisation

1. Local Curvature

Several applications within this work make use of the sensitivity function associated with the perimeter of a curve. The setting is illustrated in Fig. 9. Recall the sensitivity is defined by (3). For a curve that is represented by a set of discrete boundary points, s_α^F can be estimated as shown in Fig. 9: one displaces a boundary point α by a distance δ in the direction normal to the curve, and calculates the change in F , which in this case is the perimeter. It is convenient also to displace the boundary point by $-\delta$, which yields a central difference estimate of the sensitivity: for a boundary point with index B , this estimate is

$$s_B^F = \frac{1}{2\ell_B\delta} [F(\mathbf{X}_1, \dots, \mathbf{X}_B + \mathbf{n}_\alpha\delta, \dots, \mathbf{X}_n) - F(\mathbf{X}_1, \dots, \mathbf{X}_B - \mathbf{n}_\alpha\delta, \dots, \mathbf{X}_n)] \quad (\text{A1})$$

where ℓ_B is the length of the boundary segment associated with point B , and $F(\mathbf{X}_1, \dots, \mathbf{X}_n)$ is the value of the objective function given the boundary point positions $(\mathbf{X}_1, \dots, \mathbf{X}_n)$.

When the function F is the perimeter of the curve, the sensitivity can be calculated as shown in Fig. 9b. Using basic 2d geometry, one sees that this sensitivity can alternatively be obtained using the formula shown in Fig. 9a, and hence that this sensitivity is equal to the curvature $\kappa_B = \pm 1/R$, where R is the radius of the circle shown in Fig. 9(a): see also Ref. [4]. The sign of κ_B depends on the direction of the (inward) normal \mathbf{n} at the point B : for a (locally) convex shape then $\kappa_B < 0$ while for concave shapes $\kappa_B > 0$.

It follows that for deterministic minimisation of shape perimeter in the absence of any constraint, one should recover curvature-driven flow of the boundary Γ (effects of stochastic noise on this process have been considered by Souganidis and Yip [22]). In our scheme, the computational implementation of this process is affected by the discretisation of the level-set field, and use of discrete boundary points. In particular, this discretisation affects our estimation of sensitivities [4]. In Fig. 9(c), we initialise the level-set field as the signed distance from a circle of radius R with $\phi > 0$ outside the circle. That is, the shape Ω includes the whole of Ω_d , except for a single circular hole, as in Fig. 1. We then infer the boundary points (as described in Sec. IIB) and we measure the curvature κ_α at each such point, using the two methods shown in Fig. 9(a,b). The normal vectors \mathbf{n}_α point radially outwards on the perimeter of the circular hole, so the shape Ω is concave and one expects $\kappa_\alpha = 1/R > 0$. The results in Fig. 9(c) show the mean curvature $\bar{\kappa} = (1/n) \sum_{\alpha=1}^n \kappa_\alpha$. Error bars (for selected values of R) indicate the standard deviation $\sigma_\kappa = \sqrt{(1/n) \sum_{\alpha=1}^n (\kappa_\alpha - \bar{\kappa})^2}$. If the boundary points

were all located exactly on a circle of radius R then one would have $\kappa_\alpha = 1/R$ for all α . In practice, deriving the boundary points by linear interpolation from the level-set field leads to boundary points that are systematically slightly inside the circle, with offsets of order a_0^2/R that depend on their position on the circle (recall that a_0 is the mesh size for the spatial discretisation of the level set field). This leads to a small systematic error in the measured mean curvature $\bar{\kappa}$ [relative error of order $(a_0/R)^2$] and a standard deviation σ_κ that is approximately $\kappa/3$, independent of R (see Fig. 9c). See also Fig. 10, which is discussed below.

2. Level-set reinitialisation and its effects on curvature measurements

In Fig. 10 we investigate curvature measurements in more detail. In particular, we consider another discretisation effect, which arises from level-set reinitialisation. Recall from Sec. IID that the level-set variables are periodically reinitialised, to maintain the property that ϕ_i is the signed distance of node i from the boundary Γ . The reinitialisation procedure calculates the level-set variables ϕ_i as a function of the set of boundary point positions $\{\mathbf{X}_\alpha\}$. However, if one starts with a set of boundary points, reinitialises the level set variables, and then recalculates a new set of boundary point positions $\{\mathbf{X}'_\alpha\}$, one does not have in general $\mathbf{X}_\alpha = \mathbf{X}'_\alpha$: that is, the boundary points are changed by the reinitialisation.

This motivates the following numerical experiment, whose results are shown in Fig. 10(a). Start with a level set that encodes perfectly the signed distance from a circle of radius R . Calculate the boundary points \mathbf{X}_α and their associated local curvatures κ_α . Then reinitialise the level set based on the boundary points \mathbf{X}_α . Based on this new level set, calculate a new set of boundary points \mathbf{X}'_α and their associated curvatures κ'_α . In the absence of discretisation errors, one would have $\kappa_\alpha = \kappa'_\alpha = 1/R$ for all α . Fig. 10(a) shows averages of the κ_α and the κ'_α , evaluated by summing over boundary points and dividing by the number of boundary points. One has from Fig. 9c that the average $\bar{\kappa}$ is indeed close to $1/R$, but the average $\bar{\kappa}'$ shows significant numerical errors (deviations from $1/R$), which may be either positive or negative.

It was noted in the previous section that the estimated curvature varies with the position of the boundary point around the circle. Fig. 10(b) shows a representative set of curvature measurements (κ_α and κ'_α), for a circle of radius $R = 40$ (measured in units of a_0). When the level set field is exactly equal to the signed distance from the circle, one sees variation in the curvatures κ_α that are of the same order as κ itself, consistent with the error bars in Fig. 9c. After reinitialisation of the level set field, one sees much larger errors. The range of the κ'_α is large enough in this case that some values are even negative. The source of this error is that reinitialisation leads to movements in the boundary points of order 1–5% of a grid spacing. The

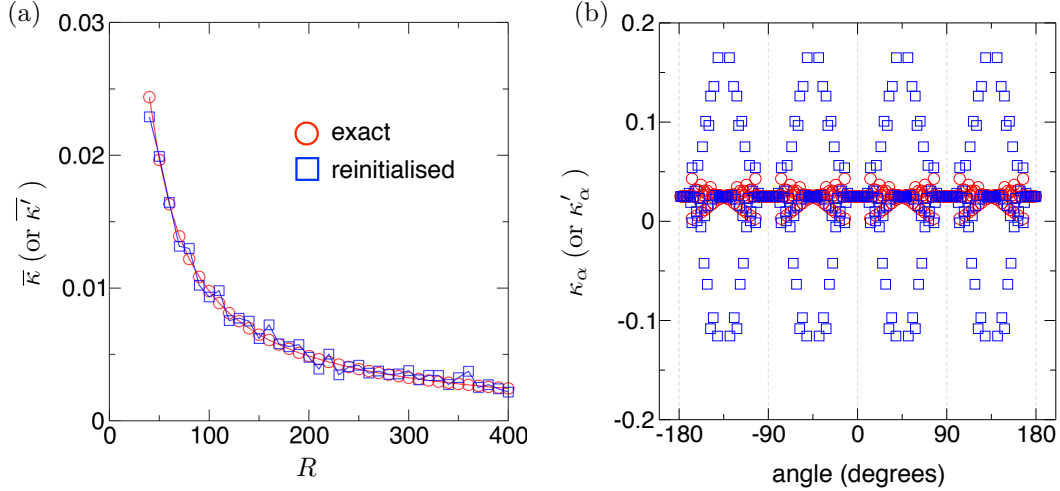


FIG. 10: Discretisation errors affect mean and local boundary point curvature. (a) Signed distance reinitialisation using a second-order Fast-Marching Method introduces errors into the mean boundary point curvature (blue squares). When the nodes of the level set domain are initialised using an exact signed-distance function (using the exact Euclidean distance to the interface) the mean curvature agrees near perfectly with the analytical result (red circles). In this case the only errors are introduced by the piece-wise linear discretisation of the boundary. (b) Local boundary point curvature around a circle of radius $R = 40$. The angle is measured relative to the top of the circle. While the mean curvature is excellent, noise is present in the local boundary curvature, even when using a perfect signed-distance function (red circles). Reinitialisation of the signed-distance function leads to a significant increase in the noise (blue squares).

typical size of these movements depends weakly on the circle radius. This leads to a standard deviation $\sigma_{\kappa'}$ that is approximately 0.06, which is larger than the curvature itself, in contrast to $\sigma_\kappa \approx 0.008 \approx \kappa/3$ when the level set is equal to the exact signed distance function.

3. Interpretation

There are two conclusions from this analysis. First, even if the level-set function is an accurate description of a shape (as with the analytic signed distance function considered here), one expects uncertainties in sensitivities, due to discretisation errors. This can influence the

convergence of the deterministic method to a minimum of F , and the extent to which the stochastic method samples (14). Accurate estimation of sensitivities is therefore an important part of any future application of this method. Second, reinitialisation of the level set can lead to small but significant movements of boundary points, which are large enough that local sensitivities change considerably. In deterministic optimisation, the frequency of reinitialisation reduces as the system converges to the optimum, and this effect is not too pronounced. On the other hand, in stochastic optimisation, reinitialisation is more frequent, and can affect the shapes Ω generated by the method. A more detailed analysis of these effects will be included in future work.

-
- [1] G. Allaire, F. Jouve, and A.-M. Toader. Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics*, 194(1):363 – 393, 2004.
 - [2] M. P. Bendsoe and O. Sigmund. *Topology optimization: theory, methods, and applications*. Springer Science and Business Media, 2013.
 - [3] C. J. Brampton, P. D. Dunning, and H. A. Kim. Stress constrained optimization using slp level set topology optimization method. *AIAA SciTech Conference*, Jan 2015.
 - [4] E. Calabi, P. J. Olver, C. Shakiban, A. Tannenbaum, and S. Haker. Differential and numerically invariant signature curves applied to object recognition. *International Journal of Computer Vision*, 26(2):107–135, 1998.
 - [5] D. Chandler. *Introduction to modern statistical mechanics*. Oxford University Press, New York, Oxford, 1987.
 - [6] J. D. Deaton and R. V. Grandhi. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization*, 49(1):1–38, 2014.
 - [7] P. D. Dunning and H. Alicia Kim. A new hole insertion method for level set based structural topology optimization. *Int. J. Numer. Meth. Eng.*, 93(1):118–134, 2013.
 - [8] P. D. Dunning and H. A. Kim. Introducing the sequential linear programming level-set method for topology optimization. *Structural and Multidisciplinary Optimization*, 51(3):631–643, 2015.
 - [9] P. D. Dunning, H. A. Kim, and G. Mullineux. Investiga-

- tion and improvement of sensitivity computation using the area-fraction weighted fixed grid fem and structural optimization. *Finite Elements in Analysis and Design*, 47(8):933–941, 8 2011.
- [10] P. D. Dunning, B. K. Stanford, and H. A. Kim. Coupled aerostructural topology optimization using a level set method for 3d aircraft wings. *Structural and Multidisciplinary Optimization*, 51(5):1113–1132, 2015.
- [11] D. Frenkel and B. Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. Elsevier, 2001.
- [12] O. Juan, R. Keriven, and G. Postelnicu. Stochastic motion and the level set method in computer vision: Stochastic active contours. *Int. J. Comp. Vis.*, 69(1):7–25, 2006.
- [13] A. Kasaiezadeh and A. Khajepour. Active contours with stochastic fronts and mechanical topology optimization. *Journal of Mechanical Design*, 135(4):041008–041008, 03 2013.
- [14] G.-C. Luh, C.-Y. Lin, and Y.-S. Lin. A binary particle swarm optimization for continuum structural topology optimization. *Applied Soft Computing*, 11(2):2833 – 2844, 2011. The Impact of Soft Computing for the Progress of Artificial Intelligence.
- [15] A. G. M. Michell. LVIII. the limits of economy of material in frame-structures. *Phil. Mag. (Series 6)*, 8(47):589–597, 1904.
- [16] B. K. Oksendal. *Stochastic differential equations : an introduction with applications*. Universitext. Springer, Berlin, Heidelberg, New York, 1998.
- [17] S. Osher and R. P. Fedkiw. *Level set methods and dynamic implicit surfaces*. Applied mathematical science. Springer, New York, N.Y., 2003.
- [18] E. Rosta and G. Hummer. Error and efficiency of replica exchange molecular dynamics simulations. *J. Chem. Phys.*, 131(16):165102, 2009.
- [19] O. Sigmund. Design of multiphysics actuators using topology optimization – part i: One-material structures. *Computer Methods in Applied Mechanics and Engineering*, 190(49–50):6577 – 6604, 2001.
- [20] O. Sigmund. On the usefulness of non-gradient approaches in topology optimization. *Structural and Multidisciplinary Optimization*, 43(5):589–596, 2011.
- [21] R. Sivapuram, P. D. Dunning, and H. A. Kim. Simultaneous material and structural optimization by multiscale topology optimization. *Structural and Multidisciplinary Optimization*, 2016. (online at <http://doi.org/10.1007/s00158-016-1519-x>).
- [22] P. Souganidis and N. Yip. Uniqueness of motion by mean curvature perturbed by stochastic noise. *Annales de l’Institut Henri Poincaré (C) Non Linear Analysis*, 21(1):1 – 23, 2004.
- [23] Y. Sugita and Y. Okamoto. Replica-exchange molecular dynamics method for protein folding. *Chem. Phys. Lett.*, 314(1–2):141 – 151, 1999.
- [24] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’94, pages 311–318, New York, NY, USA, 1994. ACM.
- [25] S. Wang and K. Tai. Structural topology design optimization using genetic algorithms with a bit-array representation. *Computer Methods in Applied Mechanics and Engineering*, 194(36–38):3749 – 3770, 2005.
- [26] C.-Y. Wu and K.-Y. Tseng. Topology optimization of structures using modified binary differential evolution. *Structural and Multidisciplinary Optimization*, 42(6):939–953, 2010.
- [27] S. Zhou, H. Sun, L.-T. Cheng, J. Dzubiella, B. Li, and J. A. McCammon. Stochastic level-set variational implicit-solvent approach to solute-solvent interfacial fluctuations. *J. Chem. Phys.*, 145(5):054114, 2016.
- [28] <http://ab-initio.mit.edu/wiki/index.php/nlopt>.
- [29] <http://github.com/lohedges/slsim>.